UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/852,351	05/08/2001	G. Christian Alford	004939.P005	4684

5073 7590 06/09/2004

BAKER BOTTS L.L.P.
2001 ROSS AVENUE
SUITE 600
DALLAS, TX 75201-2980

EXAMINER	
BANANKHAH, MAJID A	
ART UNIT	**PAPER NUMBER**
2127	

DATE MAILED: 06/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/852,351	ALFORD ET AL.	
	Examiner	Art Unit	
	Majid A Banankhah	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 May 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *3-18-02, 9-13-02,*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to application filed on May 08, 2001. Claims 1-31 are considered for examination.

2. Claim 17 depend on claim 17 and therefore is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-31 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1-27 of copending Application No.09/792,550 (hereinafter '550'). Although the conflicting claims are not identical, they are not patentably distinct from each other. For example, claim 1 of the present application is an apparatus claim that has same limitations as recited in claim 1 of the '550' application except, claim 1 in the present application recite "preemptive" thread and "cooperative" thread and scheduling the tasks and threads based on "preemptive" thread's and "cooperative" thread's priority level. However, classifying tasks as "cooperative" and "preemptive" and scheduling tasks based on priority level in this environment is well known in the art as Farrell evidences it. The reference of Farrell teaches of thread environment where threads are organized into class. Threads from the same class are not preemptable by any thread from the same class (cooperative threads) but can be preempted by a thread from a different class (preemptive threads, see Farrell, Fig, 1, col. 2, line 44 to col. 3, line 36), for the reason to permit an application program developer to influence the order of execution, and t gain greater control by the application program developer (see Farrell, col. 2, lines 21-40). Therefore, it would have been obvious for one

ordinary skill in the art at the time the invention was made to use thread scheduling technique of Farrell, in order to have more control over the threads and increase efficiency by doing that.

As another example, claim 9 in the present application recite the same limitation as the one recited in claim 8 in the '550' application, where, except, the message passing between tasks for the purpose of communication is added to claim 9 in the present application. However, it is well known in the art to use messages between tasks in order to provide communication between tasks in one and or two different environment. This would make the system more efficient because it will create more control over scheduling and the system behaves more efficiently.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 103

5. Following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith (U.S. Pat. No. 5,421,013, hereinafter Smith), in view of Alfieri (U.S. Pat. No. 5,421,013, hereinafter Alfieri) and further in view of Farrell (EP 0527 392 A2, hereinafter, Farrell).

Per claims 1 and 20, a portable thread environment comprising (Smith, portable multithreading application, and independent of platform operating system):

an application programming interface configured to support multiple application program tasks with each task comprised of one or more threads (creating application program using API, col. 2, lines 39-47);

host adaptation logic (application programming interface agency, col. 3, lines 23-39) for communicatively interfacing said tasks and threads (API agency, multithreaded application programs that use the agency, and suspending activated threads, col. 3, lines 23-39) with a host-processing environment (new platform, col. 3, lines 23-39).

Smith fails to explicitly teach of a scheduler to determine an execution order of the threads based on execution variables. However, the reference of Alfieri, in the same field of endeavor teaches of a scheduling method in a multithreading environment, wherein the threads are scheduled based on the threads priority and scheduling parameters (see Alfieri, Abstract, and col. 1 lines 58

to col. 2, line 17) for the reason to be able to execute threads in a orderly fashion and give every thread a chance to be executed based on their urgency.

The reference of Smith fails to further explicitly teach of "cooperative" task and "preemptive" task and scheduling the tasks and threads based on cooperative threads and preemptive thread's priority level. However, Farrell in the same field of endeavor teaches of thread environment where threads are organized into class. Threads from the same class are not preemptable by any thread from the same class (cooperative threads) but can be preempted by a thread from a different class (preemptive threads, see Farrell, Fig, 1, col. 2, line 44 to col. 3, line 36), for the reason to permit an application program developer to influence the order of execution, and to gain greater control by the application program developer (see Farrell, col. 2, lines 21-40). Therefore, it would have been obvious for one ordinary skill in the art at the time the invention was made to use thread scheduling technique of Farrell, in order to have more control over the threads and increase efficiency by doing that.

Per claim 2, the portable thread environment as in claim 1 wherein threads of the same preemptive task or cooperative task have different priority levels (see Farrell, col. 5, lines 39-58, priority level in each dispatch queue, and different dispatch queue).

Per claim 3, the portable thread environment as in claim 1 wherein a currently running preemptive thread is suspended by a higher priority thread (Farrell, 6, lines 29-36, thread is suspended).

Per claim 4, the portable thread environment as in claim 1 wherein a preemptive thread may suspend a currently running lower priority thread (Farrell, preemptive threads, see Farrell, Fig, 1, col. 2, line 44 to col. 3, line 36, when the running thread from one class is preempted by a higher priority thread from another dispatch class).

Per claim 5, the portable thread environment as in claim 1 wherein a currently running cooperative thread is suspended when a preemptive higher priority thread is requested (Farrell, col. 5, line 39 to col. 6, line 36).

Per claim 6, the portable thread environment as in claim 1 wherein a currently running cooperative thread of a task is not suspended when a requested cooperative thread of the task is requested (In Farrell, a thread from the same dispatch class cannot preempt another running thread, see col. 2 Summary of the Invention).

Per claim 7, the portable thread environment as in claim 1 wherein a currently running cooperative thread of a first task is suspended when a requested cooperative thread of a second task and having a higher priority level is requested (See Farrell, col. 11, line16 to col. 12, line 7, thread suspend condition, and).

Per claim 8, the portable thread environment as in claim 1 wherein a first thread may change a priority level of a second thread (col. 7, line 15 to col. 8 line 17, one thread changes thread's priority of another when highest priority completes it execution and the other one becomes the highest priority thread).

Per claims 9 a method for porting an application from a first host environment to a second host environment, said second host environment having application specific hardware comprising (Smith, Abstract, and col. 1, lines 66 to col. 2, line 2, col. 3, lines 40-59, Windows, and OS/2):

removing said first set of tasks from said application (application programming interface agency, col. 3, lines 23-39); and loading said application without said first set tasks to said second host environment (API agency, multithreaded application programs that use the agency, and suspending activated threads, col. 3, lines 23-39), said application-specific hardware providing said functions provided by said first set of tasks in said first host environment; and (new platform, col. 3, lines 23-39)

configuring said first set of tasks and second set of tasks to communicate by passing a set of messages in said first host environment, wherein one or more of said set of messages are also used to provide communication between said first set of tasks and said application-specific hardware in said second host environment (Smith, col. 2, lines 57 to col. 3 line 2, and line 23-39).

Smith fails to explicitly teach of modeling functions supported by said application-specific hardware in the form of a first set of tasks within said application. However, the reference of Alfieri, in the same field of endeavor teaches of a scheduling method [modeling function] in a multithreading environment, wherein the threads are scheduled based on the threads priority and scheduling parameters (see Alfieri, Abstract, and col. 1, lines 58 to 68) for he reason to be able to execute the tasks in a orderly fashion and give every task a chance to be executed according to a scheduling method.

Per claims 10, 17 and 25, the method as in claim 9 further comprising: interrupting the first set of tasks. By definition interrupt is the act of suspending microprocessors operation , saving the status of its work and transfer control to another thread, and this is inherent in Farrell (col. 8, lines 18-40).

Per claims 11, 18 and 28, the method as in claim 10, wherein interrupting the first set of tasks further comprises:
determine an execution order of cooperative threads and preemptive threads based on each cooperative thread's and preemptive thread's priority levels. Please see the rejection of claim 1, last paragraph.

Per claims 12, 14, 19 and 27, the preemption does not happen when the priority of the highest priority thread in other dispatch class is not higher than the priority of the running thread (Farrell, col. 5, lines 39-58).

Per claim 13, the method as in claim 11, further comprising: suspending a currently running cooperative thread when a preemptive thread is requested. See the rejection of claim 5 above.

Per claim 15, the method as in claim 11, further comprising: suspending a currently running cooperative thread of a first task when a requested cooperative thread of a second task and having a higher priority level is requested. See Farrell in col. 11, lines 16 to col. 12, line 7.

Per claims 16 and 24, see the rejection of claim 9 above.

Per claims 20 and 28, see different preemption scheduling method in the background of Farrell (col. 10, lines 16-61, when the dispatch class has no current thread in the run list.

Per claims 21 and 29, see the rejection of claim 3 above.

Per claims 22 and 30, the preemption in Farrell happens for a higher priority thread from a different class (preemptive) and not the same class (cooperative).

Per claims 23 and 31, the method of preemption of thread and tasks are the same as a task can be a combination of a few threads. Therefore, it is obvious to use preemption method of Farrell foe a task for the reason to increase efficiency when time between interruptions is more than the time between threads interruption .

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Majid A. Banankhah** whose voice telephone number is **(703) 308-6903**. A voice mail service is also available at this number.

All response sent to U.S. Mail should be mailed to:
Commissioner of Patent and Trademarks
Washington, D.C. 20231

Hand-delivered responses should be brought to Crystal Park Two, 2021 Crystal Drive, Arlington. VA, Sixth Floor (Receptionist). All hand-delivered responses will be handled and entered by the docketing personnel. Please do not hand deliver responses to the Examiner.

All Formal or Official Faxes must be signed and sent to either (703) 308-9051 or (703) 308-9052. Official faxes will be handled and entered by the docketing personnel. The date of entry will correspond to the actual FAX reception date unless that date is a Saturday, Sunday, or a Federal Holiday within the District of Columbia, in which case the official date of receipt will be the next business day. The application file will be promptly forwarded to the Examiner unless the application file must be sent to another area of the office, e.g., Finance Division for fee charging, etc.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is **(703) 305-9600**.

Majid Banankhah
6/6/04

MAJID A. BANANKHAH
PRIMARY EXAMINER